



INFORMATICA STUDENTENWERKING

Linux Workshop

Shell-ologie 101



INFORMATICA STUDENTENWERKING

Snel op weg met de command line

```
> echo | openssl s_client -connect iswleuven.be:443 2>/dev/null
```

pipe argument flag Redirect

command

- Command
 - Programma dat we runnen
- Pipe
 - “De output van het commando naar een ander commando”
- Arguments and flags
 - Opties van de command
- Redirects
 - “Output van een commando naar een file of stream”

Shell-ologie – Pipes vs redirects

```
> cat stuff.txt | grep 'Annie'
Annie
```

- Pipes (|)
 - Output van cmd1 naar Input cmd2
 - **Belangrijk voor one-liners!**
 - Alles in STDOUT gaat naar STDIN
- Een pipe is shorthand voor:
 - thing1 > temp_file && thing2 < temp_file

```
> grep -v 'Annie' oops.txt 2>./error
> cat error
grep: oops.txt: No such file or directory
>
```

- Redirects (> en >>)
 - Output van cmd1 naar file of stream
 - in dit geval naar file “./error”
 - Output kan je kiezen
 - ≥ is STDOUT
 - 2≥ is STERR (errors)
 - 2>&1 is voor alles
 - >> is voor appending
 - Handig om errors weg te werken
 - /dev/null is letterlijk “niks” in linux!

```
> grep -v 'Annie' oops.txt 2>/dev/null
>
```

Shell-ologie - Looping

```
> for i in {1..3}; do echo "Hi, $i"; done
Hi, 1
Hi, 2
Hi, 3
```

- Zoals in Python
 - **for** var name in *stuff* ;
 - **do** *stuff* \$var name ;
 - **done**
- *stuff* kan
 - Explicit zijn (-> for i in Marie Bob Annie;)
 - Commandos zijn (-> for i in \$(cat stuff.txt);)
 - Spread operators zijn (voorbeeld)
- Sidenotes;
 - \$var is een var
 - ; stopt een 'lijn'

Shell-ologie - Variables

```
> the_time=$(date +%H:%M:%S)
> echo $the_time
16:54:45
```

- Var=value
- **Let op!**
 - Als je een cmd output wilt opslaan, **gebruik \$(cmd)**
 - Handig om te weten voor later
- **\$(cmd) kan je bijna overal gebruiken!**

Shell-ologie – if, elif, else

```
> num=42; if [ $num -eq 42 ]; then echo "The number is 42"; elif [ $num -gt 42 ]; then echo "The number is greater than 42"; else echo "The number is less than 42"; fi
The number is 42
> num=41; if [ $num -eq 42 ]; then echo "The number is 42"; elif [ $num -gt 42 ]; then echo "The number is greater than 42"; else echo "The number is less than 42"; fi
The number is less than 42
> num=43; if [ $num -eq 42 ]; then echo "The number is 42"; elif [ $num -gt 42 ]; then echo "The number is greater than 42"; else echo "The number is less than 42"; fi
The number is greater than 42
```

- `if [ test in here ]; then cmd; else cmd; fi`
 - Old but trusty method
 - `-eq` => equals
 - `-lt` => less than
 - `-gt` => greater than
- `If [[ $name == "Bob" ]]`
 - Nieuwe versie
 - Compatiebel met de `[ ]` versie
 - Maar laat string comparisons toe

- We hebben 100 files
 - 99 daarin zeggen dat Annie niet hier is
 - 1 bevat letterlijk alleen "Annie"
- Vind de file waar er alleen Annie in staat
 - De output zou dit zijn
 - `/home/terminator/tmp/labo/ex1/findme4.txt`

```
> printf '%s\n' file1 file2 file3 | touch
touch: missing file operand
Try 'touch --help' for more information.
> printf '%s\n' file1 file2 file3 | xargs touch
> ls file?
file1  file2  file3
```

- Soms werken pipes niet
 - Too many arguments...
 - Meerdere outputs..
- Oplossing -> **xargs**
 - command1 | xargs command2
 - Best te gebruiken om commandos te bouwen!

Shell-ologie - Oefening

- Pas je vorige oneliner
 - Print de informatie van de file met commando **stat**
- Voorbeeld:

```
> stat findme1.txt
File: findme1.txt
Size: 18          Blocks: 8          IO Block: 4096   regular file
Device: 8,0      Inode: 303132      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/terminator)   Gid: ( 1000/terminator)
Access: 2023-11-27 17:26:42.197841267 +0100
Modify: 2023-11-27 17:26:30.947841203 +0100
Change: 2023-11-27 17:26:30.947841203 +0100
Birth: 2023-11-27 17:23:19.737840106 +0100
```

Shell-ologie – Extra oefenen?

- Gebruik zo veel mogelijk de shell
 - Bestuur Docker via CLI
 - Git bash gebruiken
- Check de uclllabs wiki (wiki.uclllabs.be)
 - Extra oefeningen (en oplossingen) staan daar
- Maak je eigen oefeningen

Shell-ologie – Afsluitende tips

- Spam **tab** zo veel mogelijk
 - Autocomplete geeft je een idee wat je kan doen
- Gebruik de **apropos** command
 - “apropos zip file” geeft je alle commandos over zip files
- Gebruik het **man** command
 - Shorthand voor manual
 - Uitleg over het commando
 - Gebruik / om het door te zoeken
- Keep it simple!

Shell Speedrunning tricks



INFORMATICA STUDENTENWERKING

Sneller werken met de shell

Shell tricks – Bang Bangs

- Scenario; snel iets installeren
 - apt install thefuck
 - damn, sudo vergeten
- Gebruik **sudo !!**
 - !! -> **vorige commando**
- Deze werken ook
 - !s -> vorig commando dat begint met s
 - !-10 -> de 10^{de} vorige commando
 - !!* -> de parameters van de vorige commando
 - !!:p -> print het vorig commando

- **Alt-f/b**
 - Cursor naar volgende/vorige woord
- **Control-a/e**
 - Cursor naar begin/einde van lijn
- **Control-xx**
 - Cursor swappen tussen twee plekken

Shell tricks – Kill and Yank

- Beter gekend als de cut en paste
 - Een snellere backspace
- Control+K -> van **cursor** tot **einde van de lijn**
- Control+U -> van **cursor** tot **begin van de lijn**
 - **Zsh** -> Alt+w (ctrl+u verwijderd lijn)
- Control+W -> cut woord
- Control+Y -> de tekst plakken

Shell tricks – Searching

- Scenario; je wilt een one liner terug vinden van vorige week

- Beste manier? **bck-i-search**

```
terminator@tmper ➤ ~/tmp/lab0/ex1 ➤ echo $the_time  
bck-i-search: the__
```

- Start zoeken met **Control+R**

- Typ een stuk om te zoeken
- Niks gevonden? Opnieuw **Control+R** voor de volgende result.
- Te ver? Gebruik **Control+S** om terug te gaan
- Gevonden? **Enter** om te bevestigen
- Opgeven? **Control+C**

```
terminator@tmper ➤ ~/tmp/lab0/ex1 ➤ the_time=$(date +%H:%M:%S)  
fwd-i-search: the__
```

Shell tricks – Editor voor shell

- Een lange oneliner aan het schrijven?
- Druk **Control+X Control+E** voor een editor
 - Makkelijker om overzichtelijk te werken
 - Gebruikt whatever \$EDITOR is
 - Voor VIM:
 - **I** om te typen
 - **Escape -> :qa** om te exiten
- Oneliner afgewerkt?
 - Save de file om uit te voeren

```
terminator@tmper ➤ ~/tmp/lab0/ex1 ➤ for file in ~/tmp/lab0/ex1/findme*.txt;  
do echo $file;  
done;
```

Shell tricks – Snelle Afsluiters

- **Globs**
 - Selecteer meerdere files
 - E.g **findme*.txt**
 - Alles wat begint met **findme** en eindigt met **.txt**
- **Control+C** om rap je lijn te clearen
- **cd** – om naar de vorige directory te gaan
- **Aliases**
 - Vervang **ls -lah** door **l**

Know your (command line) tools



INFORMATICA STUDENTENWERKING

```
> ps -aux | grep 'rpc'
rpc          20841  0.0  0.0   8076  4024 ?        Ss   Nov27   0:00 /usr/bin/rpcbind -w -f
rpcuser      20842  0.0  0.0   4328  1940 ?        Ss   Nov27   0:00 /usr/sbin/rpc.statd
```

- Woorden zoeken
 - In files of pipes
 - Met regex
- **Enorm** handig!
- Snelle oefening
 - Herschrijf de eerste oefening zodat je alleen grep gebruikt
 - Tip, gebruik **globs**!
 - Voorbeeld van uitput: `findme39.txt:Annie`

- nano
 - Makkelijk om mee te starten
 - Makkelijk om te masteren
 - Niet *zo* advanced
- Shortcuts lezen
 - **^** -> **Control**
 - **M** -> **Alt/Option**
 - Voorbeeld
 - **Control+w** -> Zoeken in file
 - Protip: verder zoeken door **^w** en **Enter** te drukken
 - **Alt+u** -> Undo